

ISIS-II PL/M-86 V2.0 COMPILATION OF MODULE PREDEF

OBJECT MODULE PLACED IN PREDEF.OBJ

COMPILER INVOKED BY: :F0: PREDEF.PL M DEBUG OPTIMIZE(2) DATE(FEBRUARY 2, 1982) XREF

```

1      title ('Predefined symbols')
      predef:
      do;

/*

      modified 7/24/81 R. Silberstein

*/

/***** Module for predefined symbols *****/

/*
      This module contains the tables and subroutines
      for the PREDEFINED symbols of the ASM86 assembler.

      The subroutine

          PFIND (nochar,stringaddr,attributeaddr) byte

      defines the interface to the other modules.

      The routine tests if a given symbol is a predefined
      symbol. If so the address of the symbol attributes
      is returned.
      The format of the symbol attributes is :

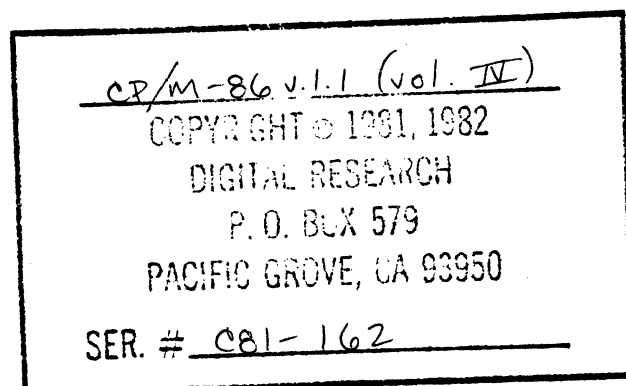
          byte
          0  * symbol type *
          1  * symbol description *
          2  *
          3  * symbol value *
          ****

/*      Include language macros      */
/*      and general assembler        */
/*      definitions (literals).      */

#include(:f1:macro.lit)
= $nolist
#include(:f1:equals.lit)
= $nolist

/*      Predefined numbers:      */

```



```

3 1  dcl
      nbyte lit '1',
      nword lit '2',
      ndword lit '4';

```

JAN 10 1982
 DIGITAL RESEARCH
 P.O. BOX 579
 PACIFIC GROVE, CA 93950
 SER. # _____

```
$eject
/*
```

Here are the predefined symbols of
the assembler.

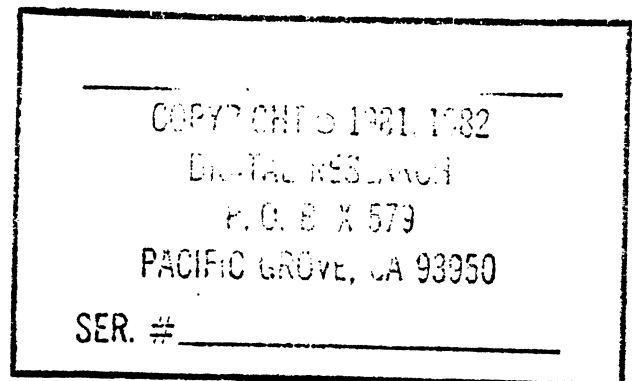
The symbols are grouped according to
the symbol lengths. Moreover, each group
is sorted alphabetically so that binary
search algorithm could be used.

```
*/
```

```
9 1 dcl
tok0(*) byte data(0), /* symbollength > 1 only */
tok1(*) byte data(0),
tok2(*) byte data(34,'AHALAXBHBLBPBXCHLCSCXDBDDDDHDIDL',
'DSDWDXEQESGEGTIFLELTNEORRBRSRWSISPSS'),
tok3(*) byte data(11,'ANDENDEQUODNOTORGPTRSEGSLSHRXOR'),
tok4(*) byte data(13,'BYTECSEGBITDSEGENDESEGLASTLISTRELBRELW',
'SSEGTPEWORD'),
tok5(*) byte data(6,'DWORDEJECTENDIFMODRMSHORITITLE'),
tok6(*) byte data(5,'IFLISTLENGTHNOLISTOFFSETSEGFIX'),
tok7(*) byte data(2,'INCLUDESIMFORM'),
tok8(*) byte data(3,'NOIFLISTNOSEGFIXPAGESIZE'),
tok9(*) byte data(2,'CODEMACROPAGewidth');

/* Pointer table: */
```

```
10 1 dcl
tokpointer(*) address data(.tok0,.tok1,.tok2,.tok3,.tok4,
.tok5,.tok6,.tok7,.tok8,.tok9);
```



```
%object
```

```
/*
```

```
This is the attribute table for
the predefined symbols.
```

```
*/
```

```
11 1
```

```
dcl
```

```
value0      byte,
```

```
value1      byte,
```

```
value2(34)   struc  (type byte,descr byte,value addr) data(
    reg,byt,rah,    reg,byt,ral,    /* registers AH and AL */
    reg,wrdrax,     reg,byt,rax,    /* registers AX and BH */
    reg,byt,rbl,     reg,wrdrbp,    /* registers BL and BP */
    reg,wrdrbx,     reg,byt,rch,    /* registers BX and CH */
    reg,byt,rcl,     reg,dwdr,rcs,   /* registers CL and CS */
    reg,wrdr,rcx,    /* register CX */
    pseudo,nil,pdb,  pseudo,nil,pdd, /* pseudos DB and DD */
    reg,byt,rdh,     reg,wrdr,rdi,   /* registers DH and DI */
    reg,byt,rdl,     reg,dwdr,rds,   /* registers DL and DS */
    pseudo,nil,pdw,  /* pseudo DW */
    reg,wrdr,rdx,    /* register DX */
    operator,nil,oeq, /* operator EQ */
    reg,dwdr,res,    /* register ES */
    operator,nil,oge, /* operator GE */
    operator,nil,ogt, /* operator GT */
    pseudo,nil,pif,  /* pseudo IF */
    operator,nil,ole, /* operator LE */
    operator,nil,olt, /* operator LT */
    operator,nil,one, /* operator NE */
    operator,nil,oor, /* operator OR */
    pseudo,nil,prb,  /* pseudo RB */
    pseudo,nil,prs,  /* pseudo RS */
    pseudo,nil,prw,  /* pseudo RW */
    reg,wrdr,rsi,    reg,wrdr,rsp,   /* registers SI and SP */
    reg,dwdr,rss),   /* register SS */
```

```
value3(11)   struc  (type byte,descr byte,value addr) data(
    operator,nil,oand, /* operator AND */
    pseudo,nil,pend,pseudo,nil,pequ, /* pseudos END and EQU */
    operator,nil,omod, /* operator MOD */
    operator,nil,onot, /* operator NOT */
    pseudo,nil,porg,   /* pseudo ORG */
    operator,nil,optr,  /* operator PTR */
    operator,nil,oseg,  /* operator SEG */
    operator,nil,oshl,  /* operator SHL */
    operator,nil,oshr,  /* operator SHR */
    operator,nil,oxor), /* operator XOR */
```

```
value4(13)   struc  (type byte,descr byte,value addr) data(
    number,byt,nbyte, /* 8 bit number BYTE (1) */
    pseudo,nil,pcseg,  /* pseudo CSEG */
    pseudo,nil,pdbit,  /* pseudo DBIT */
    pseudo,nil,pdseg,  /* pseudo DSEG */
    pseudo,nil,pendm,  /* pseudo ENDM */
    pseudo,nil,peseg,  /* pseudo ESEG */
    operator,nil,olast, /* operator LAST */
    pseudo,nil,plist,  /* pseudo LIST */
    pseudo,nil,prelb,  /* pseudo RELB */
    pseudo,nil,prelw,  /* pseudo RELW */
```

COPYRIGHT 1981, 1982

DIGITAL RESEARCH

P.O. BOX 579

PACIFIC GROVE, CA 93950

SER. #

```

pseudo,nil,psseg,          /* pseudo SSEG */
operator,nil,otype,        /* operator TYPE */
number,wrđ,nword),        /* 16 bit number WORD (2) */
value5(6)    struc (type byte,descr byte,value addr) data(
number,dwrđ,ndword,        /* 32 bit number DWORD (4) */
pseudo,nil,peject,        /* pseudo EJECT */
pseudo,nil,pendif,        /* pseudo ENDIF */
pseudo,nil,pmodrm,        /* pseudo MODRM */
operator,nil,oshort,       /* operator SHORT */
pseudo,nil,ptitle),       /* pseudo TITLE */
value6(5)    struc (type byte,descr byte,value addr) data(
PSEUDO,NIL,PIFLIST,        /* PSEUDO IFLIST */
operator,nil,olength,      /* operator LENGTH */
pseudo,nil,pnolist,        /* pseudo NOLIST */
operator,nil,ooffset,      /* operator OFFSET */
pseudo,nil,psegfix),       /* pseudo SEGFIX */
value7(2)    struc (type byte,descr byte,value addr) data(
pseudo,nil,pinclude,       /* pseudo INCLUDE */
pseudo,nil,psinfrm),       /* pseudo SIMFORM */
value8(3)    struc (type byte,descr byte,value addr) data(
PSEUDO,NIL,PNOIFLIST,      /* PSEUDO NOIFLIST */
pseudo,nil,pnosegfix,      /* pseudo NOSEGFIX */
pseudo,nil,ppagesize),     /* pseudo PAGESIZE */
value9(2)    struc (type byte,descr byte,value addr) data(
pseudo,nil,pcodemacro,     /* pseudo CODEMACRO */
pseudo,nil,ppagewidth);    /* pseudo PAGEWIDTH */

```

```

/* Pointer table: */

```

```

12 1  dcl
valuepointer(*) address data(.value0,.value1,.value2,.value3,.value4,
                             .value5,.value6,.value7,.value8,.value9);

```

COPYRIGHT © 1981, 1982

DIGITAL RESEARCH

P. O. BOX 579

PACIFIC GROVE, CA 93950

SER. # _____

\$object

/* Global variables and subroutines */

```

13 1  dcl
      nochar                byte,
      stringaddr            address,
      attributeaddr         address,
      source based stringaddr (1) byte,
      dest based attributeaddr (1) byte,
      value                  address, /* pointer to attributes */
      valuebyte based value (1) byte,
      tok                    address, /* pointer to table strings */
      tokenbyte based tok    byte,
      t$lookahead            address, /* table string pointer */
      look based t$lookahead (1) byte,
      v$lookahead            address, /* attribute table pointer */
      noleft                 byte, /* no of tablestrings left */
      half                   byte, /* noleft/2 */
      i                       byte, /* counter */
      attribute$length       lit '4', /* no of bytes pr attribute */
      equal                  lit '0', /* results of stringcompares */
      greater                 lit '1',
      less                    lit '2';

```

/* Routine to compare tablestring with given symbolstring */

```

14 1  compare: proc byte;
15 2      i=0ffh;
16 2      do while (i:=i+1) < nochar;
17 3          if source(i) > look(i) then
18 3              return greater;
19 3          if source(i) < look(i) then
20 3              return less;
22 2      end$while;
22 2      return equal;
23 2  end compare;

```

COPYRIGHT © 1981, 1982
 DIGITAL RESEARCH
 P. O. BOX 579
 PACIFIC GROVE, CA 93950

SER. #

/* Recursive routine to perform binary tablesearch */

```

24 1  binsearch: proc byte reent;
25 2      if noleft = 0 then$do
27 3          return false;
28 3      else$do
30 3          half=noleft/2;
31 3          t$lookahead=tok+half*nochar;
32 3          v$lookahead=value+half*attributelength;
33 3          do case compare;

          /* equal */
34 4          do;
35 5              value=v$lookahead; /* match found, pick up attributes */
36 5              do i=0 to attributelength-1;
37 6                  dest(i)=valuebyte(i);
38 6              end$do;

```

```

39 5      return true;
40 5      end;

      /* greater */
41 4      do;
42 5      tok=t$lookahead+nochar; /* test last half of table */
43 5      value=v$lookahead+attributelength;
44 5      noleft=noleft-half-1;
45 5      return binsearch;
46 5      end;

      /* less */
47 4      do;
48 5      noleft=half; /* test first half of table */
49 5      return binsearch;
50 5      end;

51 4      end$case;
52 3      end$if;
53 2      end binsearch;

```

/*

Interface routine PFIND :

***** pfind (nochar,stringaddr,attributeaddr) byte *****

Routine to test if a given symbol is a predefined symbol.

entry: nochar = no of character in symbol
 stringaddr = address of symbol string
 attributeaddr = address to put the symbol-attributes (if found)

exit: The routine returns TRUE if symbol found, otherwise FALSE.

*/

```

54 1      pfind: proc (n,s,a) byte public;
55 2      dcl
          n      byte,
          (s,a)  address;

56 2      nochar=n; /* pick up parameters */
57 2      stringaddr=s;
58 2      attributeaddr=a;

59 2      if nochar < 10 then$do
61 3          value=valuepointer(nochar);
62 3          tok=tokpointer(nochar);
63 3          noleft=tokenbyte;
64 3          tok=tok+1;
65 3          return binsearch;
66 3      else$do
68 3          return false;

```

COPYRIGHT © 1981, 1982

DIGITAL RESEARCH

P. O. BOX 579

PACIFIC GROVE, CA 93950

SER. # _____

```
69 3      endif;
70 2      end pfind;

/***** end of module *****/
71 1      end$module predef;
```

COPYRIGHT © 1981, 1982

DIGITAL RESEARCH

P.O. BOX 579

PACIFIC GROVE, CA 93950

SER. = _____

CROSS-REFERENCE LISTING

DEFN ADDR SIZE NAME, ATTRIBUTES, AND REFERENCES

54	0004H	2	A.	WORD PARAMETER AUTOMATIC	55	58				
2			ADDR	LITERALLY	11					
13	0002H	2	ATTRIBUTEADDR. . .	WORD	13	37	58			
13			ATTRIBUTELNGTH. .	LITERALLY	32	36	43			
24	0049H	167	BINSEARCH.	PROCEDURE BYTE REENTRANT STACK=0006H				45	49	65
4			BYT.	LITERALLY	11					
3			CODE	LITERALLY						
14	0000H	73	COMPARE.	PROCEDURE BYTE STACK=0002H				33		
2			CR	LITERALLY						
2			DCL.	LITERALLY						
3			DELETEDSYMB. . . .	LITERALLY						
11	0001H	1	DESCR.	BYTE MEMBER(VALUE9)						
11	0001H	1	DESCR.	BYTE MEMBER(VALUE4)						
11	0001H	1	DESCR.	BYTE MEMBER(VALUE6)						
11	0001H	1	DESCR.	BYTE MEMBER(VALUE2)						
11	0001H	1	DESCR.	BYTE MEMBER(VALUE3)						
11	0001H	1	DESCR.	BYTE MEMBER(VALUE7)						
11	0001H	1	DESCR.	BYTE MEMBER(VALUE5)						
11	0001H	1	DESCR.	BYTE MEMBER(VALUE8)						
13	0000H	1	DEST	BYTE BASED(ATTRIBUTEADDR) ARRAY(1)				37		
3			DOUBLEDEFINED. . .	LITERALLY						
4			DWRD	LITERALLY	11					
2			ELSEDO	LITERALLY						
2			ENDCASE.	LITERALLY						
2			ENDDO.	LITERALLY						
2			ENDFOREVER	LITERALLY						
2			ENDIF.	LITERALLY						
2			ENDMODULE.	LITERALLY						
2			ENDOFFILE.	LITERALLY						
2			ENDPROC.	LITERALLY						
2			ENDWHILE	LITERALLY	20					
13			EQUAL.	LITERALLY	22					
3			ERROR.	LITERALLY						
2			FALSE.	LITERALLY	27	68				
2			FOREVER.	LITERALLY						
2			FORMFEED	LITERALLY						
13			GREATER.	LITERALLY	18					
13	0010H	1	HALF	BYTE	30	31	32	44	48	
13	0011H	1	I.	BYTE	15	16	17	19	36	37
3			IDENT.	LITERALLY						
2			INIT	LITERALLY						
3			LAB.	LITERALLY						
7			LEFTBRACKET. . . .	LITERALLY						
13			LESS	LITERALLY	20					
2			LF	LITERALLY						
2			LIT.	LITERALLY	2	3	4	5	6	7
										8
13	0000H	1	LOOK	BYTE BASED(TLOOKAHEAD) ARRAY(1)					17	19
54	0003H	1	N.	BYTE PARAMETER AUTOMATIC	55	56				
8			NBYTE.	LITERALLY	11					

8		NWORD	LITERALLY	11						
3		NEGLECTED.	LITERALLY							
4		NIL.	LITERALLY	11						
13	000EH	1 NOCHAR	BYTE	16	31	42	56	59	61	62
13	000FH	1 NOLEFT	BYTE	25	30	44	48	63		
3		NUMBER	LITERALLY	11						
8		NWORD.	LITERALLY	11						
7		OAND	LITERALLY	11						
7		OEQ.	LITERALLY	11						
7		OGE.	LITERALLY	11						
7		OGT.	LITERALLY	11						
7		OLAST.	LITERALLY	11						
7		OLE.	LITERALLY	11						
7		OLENGTH.	LITERALLY	11						
7		OLT.	LITERALLY	11						
7		OMOD	LITERALLY	11						
7		ONE.	LITERALLY	11						
7		ONOT	LITERALLY	11						
7		OOFFSET.	LITERALLY	11						
7		OOR.	LITERALLY	11						
3		OPERATOR	LITERALLY	11						
7		OPTR	LITERALLY	11						
7		OSEG	LITERALLY	11						
7		OSHL	LITERALLY	11						
7		OSHORT	LITERALLY	11						
7		OSHR	LITERALLY	11						
7		OTYPE.	LITERALLY	11						
7		OXOR	LITERALLY	11						
6		PCODEMACRO	LITERALLY	11						
6		PCSEG.	LITERALLY	11						
6		PDB.	LITERALLY	11						
6		PDBIT.	LITERALLY	11						
6		PDD.	LITERALLY	11						
6		PDSEG.	LITERALLY	11						
6		PDW.	LITERALLY	11						
6		PEJECT	LITERALLY	11						
6		PEND	LITERALLY	11						
6		PENDIF	LITERALLY	11						
6		PENDM.	LITERALLY	11						
6		PEQU	LITERALLY	11						
6		PESEG.	LITERALLY	11						
54	00F0H	72 PFIND.	PROCEDURE	BYTE	PUBLIC	STACK=0010H				
6		PIF.	LITERALLY	11						
6		PIFLIST.	LITERALLY	11						
6		PINCLUDE	LITERALLY	11						
6		PLIST.	LITERALLY	11						
6		PMODRM	LITERALLY	11						
6		PNOIFLIST.	LITERALLY	11						
6		PNOLIST.	LITERALLY	11						
6		PNOSEGFIX.	LITERALLY	11						
6		PORG	LITERALLY	11						
6		PPAGESIZE.	LITERALLY	11						
6		PPAGEWIDTH	LITERALLY	11						
6		PRB.	LITERALLY	11						
1	0000H	PREDEF	PROCEDURE	STACK=0000H						
6		PRELB.	LITERALLY	11						
6		PRELW.	LITERALLY	11						

COPYRIGHT © 1981, 1982

DIGITAL RESEARCH

P. O. BOX 579

PACIFIC GROVE, CA 93950

SER. # _____

2		PROC	LITERALLY	14	24	54
6		PRS.	LITERALLY	11		
6		PRW.	LITERALLY	11		
6		PSEGFIX.	LITERALLY	11		
3		PSEUDO	LITERALLY	11		
6		PSINFORM	LITERALLY	11		
6		PSSEG.	LITERALLY	11		
6		PTITLE	LITERALLY	11		
5		RAH.	LITERALLY	11		
5		RAL.	LITERALLY	11		
5		RAX.	LITERALLY	11		
5		RBH.	LITERALLY	11		
5		RBL.	LITERALLY	11		
5		RBP.	LITERALLY	11		
5		RBX.	LITERALLY	11		
5		RCH.	LITERALLY	11		
5		RCL.	LITERALLY	11		
5		RCS.	LITERALLY	11		
5		RCX.	LITERALLY	11		
5		RDH.	LITERALLY	11		
5		RDI.	LITERALLY	11		
5		RDL.	LITERALLY	11		
5		RDS.	LITERALLY	11		
5		RDX.	LITERALLY	11		
2		REENT.	LITERALLY	24		
3		REG.	LITERALLY	11		
5		RES.	LITERALLY	11		
7		RIGHTBRACKET	LITERALLY			
5		RSI.	LITERALLY	11		
5		RSP.	LITERALLY	11		
5		RSS.	LITERALLY	11		
54	0006H	2 S.	WORD PARAMETER AUTOMATIC	55	57	
13	0000H	1 SOURCE	BYTE BASED(StringAddr) ARRAY(1)		17	19
2		SPACE.	LITERALLY			
3		SPEC	LITERALLY			
3		STRING	LITERALLY			
13	0000H	2 STRINGADDR	WORD	13	17	19 57
2		STRUC.	LITERALLY	11		
3		SYMBOL	LITERALLY			
2		TAB.	LITERALLY			
2		THENDO	LITERALLY	25	59	
13	0008H	2 LOOKAHEAD	WORD	13	17	19 31 42
13	0006H	2 TOK.	WORD	13	31	42 62 63 64
9	0158H	1 TOK0	BYTE ARRAY(1) DATA			10
9	0159H	1 TOK1	BYTE ARRAY(1) DATA			10
9	015AH	69 TOK2	BYTE ARRAY(69) DATA			10
9	019FH	34 TOK3	BYTE ARRAY(34) DATA			10
9	01C1H	53 TOK4	BYTE ARRAY(53) DATA			10
9	01F6H	31 TOK5	BYTE ARRAY(31) DATA			10
9	0215H	31 TOK6	BYTE ARRAY(31) DATA			10
9	0234H	15 TOK7	BYTE ARRAY(15) DATA			10
9	0243H	25 TOK8	BYTE ARRAY(25) DATA			10
9	025CH	19 TOK9	BYTE ARRAY(19) DATA			10
13	0000H	1 TOKENBYTE.	BYTE BASED(TOK)	63		
10	0000H	20 TOKPOINTER	WORD ARRAY(10) DATA			62
2		TRUE	LITERALLY	39		
11	0000H	1 TYPE	BYTE MEMBER(VALUE9)			

COPYRIGHT © 1991, 1992
DIGITAL RESEARCH
P.O. BOX 578
PLANO, TEXAS 75068-0578

SER. # _____

11	0000H	1	TYPE	BYTE MEMBER(VALUE8)					
11	0000H	1	TYPE	BYTE MEMBER(VALUE7)					
11	0000H	1	TYPE	BYTE MEMBER(VALUE6)					
11	0000H	1	TYPE	BYTE MEMBER(VALUE5)					
11	0000H	1	TYPE	BYTE MEMBER(VALUE4)					
11	0000H	1	TYPE	BYTE MEMBER(VALUE3)					
11	0000H	1	TYPE	BYTE MEMBER(VALUE2)					
3			UDEFSYMB	LITERALLY					
13	0004H	2	VALUE.	WORD	13	32	35	37	43 61
11	0002H	2	VALUE.	WORD MEMBER(VALUE9)					
11	0002H	2	VALUE.	WORD MEMBER(VALUE8)					
11	0002H	2	VALUE.	WORD MEMBER(VALUE7)					
11	0002H	2	VALUE.	WORD MEMBER(VALUE6)					
11	0002H	2	VALUE.	WORD MEMBER(VALUE5)					
11	0002H	2	VALUE.	WORD MEMBER(VALUE4)					
11	0002H	2	VALUE.	WORD MEMBER(VALUE3)					
11	0002H	2	VALUE.	WORD MEMBER(VALUE2)					
11	000CH	1	VALUE0	BYTE	12				
11	000DH	1	VALUE1	BYTE	12				
11	0014H	136	VALUE2	STRUCTURE ARRAY(34) DATA					12
11	009CH	44	VALUE3	STRUCTURE ARRAY(11) DATA					12
11	00C8H	52	VALUE4	STRUCTURE ARRAY(13) DATA					12
11	00FCH	24	VALUE5	STRUCTURE ARRAY(6) DATA					12
11	0114H	20	VALUE6	STRUCTURE ARRAY(5) DATA					12
11	0128H	8	VALUE7	STRUCTURE ARRAY(2) DATA					12
11	0130H	12	VALUE8	STRUCTURE ARRAY(3) DATA					12
11	013CH	8	VALUE9	STRUCTURE ARRAY(2) DATA					12
13	0000H	1	VALUEBYTE.	BYTE BASED(VALUE) ARRAY(1)					37
12	0144H	20	VALUEPOINTER	WORD ARRAY(10) DATA					61
3			VARIABLE	LITERALLY					
13	000AH	2	VLOOKAHEAD	WORD	32	35	43		
4			WRD.	LITERALLY		11			

MODULE INFORMATION:

CODE AREA SIZE	= 0138H	312D
CONSTANT AREA SIZE	= 026FH	623D
VARIABLE AREA SIZE	= 0012H	18D
MAXIMUM STACK SIZE	= 0010H	16D
462 LINES READ		
0 PROGRAM ERROR(S)		

END OF PL/M-86 COMPILATION

